H E S P L O T

by Jay Balakrishnan

T A B L E O F C O N T E N T S

Human Engineered Software
3748 Inglewood Blvd. Room 11
Los Angeles, California 90066

1.Ø Introduction

Thank you for purchasing HESPLOT, a product of Human Engineered Software. We welcome comments and suggestions pertaining to HESPLOT. HESPLOT is a utility program that allows you to use the high-resolution (hi-res) graphics capabilities of the VIC. An 8K VIC (standard 5K VIC plus 3K memory expansion) is highly recommended, although HESPLOT can be used in a reduced capacity with a 5K VIC. HESPLOT is actually two machine language programs - VECTOR draws or erases lines between any two points and PLOT can clear the hi-res screen or set or reset any individual point (within a 176 by 16Ø dot field) on the hi-res screen. To set a point means to turn it on, make it displayable. To reset a point means to turn it off, making it invisible.

1.1 Concepts and Capabilities

The only hi-res capability possible on the VIC utilizes the user-definable characters. This is an inflexible and difficult technique. Because of this, there are a few limitations in using HESPLOT. First, you can do hi-res plotting in only one color. It can be any one of the eight foreground colors. HESPLOT will use the current foregound color. Second, because of the memory requirements of both the HESPLOT program and the hi-res area, an 8K VIC is needed to fully utilize the hi-res graphics. Finally, because of the addressing scheme of the VIC 656Ø chip, HESPLOT can ONLY be used on a VIC with maximum of 8K of memory. If you expand it beyond that, the screen and color mapping areas in the VIC are moved around, and conflicts with the hi-res area. Therefore, when using HESPLOT, remove all memory expansion catridges that you may have except for the 3K expansion, which should stay plugged in.

You do not really need to understand about definable characters to use HESPLOT, but if you do want to learn more about it, see the reference section (3.1) for information about it. What you do need to know is that using the hi-res option uses memory, quite a lot of it. This memory must come from the user memory that the VIC has. That means the larger the area that you use for hi-res graphics, the less you have for BASIC programs, and vice-versa. HESPLOT has the capability to set/reset a maximum of 176 dots horizontally by 16Ø dots vertically, for a total of 28160 dots. You need an 8K VIC for that. Because HESPLOT is not too useful on a 5K VIC, we will emphasize 8K VIC examples. You will always be able to display 176 dots (or pixels, short for "picture elements") across. But unless you have 8K, you will have less than 16Ø pixels vertically.

When you use HESPLOT, you must decide how much of the memory will be used by your program and how much will be used by the hi-res graphics memory. Accordingly, you should partition memory and set aside memory for the hi-res area. Whatever is left will be used by your BASIC program. The table below will help you with this.

TABLE 1 - Hi-res Graphics Memory Usage Table

Mode	POKE 36869	Starting Mem. Loc.	POKE 56	Bytes Used	Vert. Resol.	No. of Chars.

1.	255	7168	28	512	16	1*64=64
2.	254	6144	24	1536	64	3*64=192
3.	253	5120	20	2560	112	5*64=320
4.	252	4096	16	3584	160	7*64=448

The second column, "POKE 36869" contains different numbers that will be POKED to setup different sizes of hi-res graphic area. "Starting Mem. Loc." specifies the starting byte of the hi-res graphics area that will be in use if the number in the second column is POKED into location 36869. "POKE 56" column contains the number that should be POKEd into location 56 in order to set aside and protect the hi-res area. "Bytes used" indicates the number of bytes that will be used for the hi-res graphic area. This has to come out of the total amount of memory available. "Vert. Resol." shows the vertical resolution, which is the number of pixels that you can plot vertically. You can always plot 176 horizontally. Vertical plotting always starts from the top of the screen. The origin (the origin is where the X-coordinate and Y-coordinate are both 0) is at the upper-left corner of the screen. Finally, "No. of Chars." shows the number of characters that are being redefined. It is a multiple of 64.

We will now show you how to use this table to decide what configuration of hi-res graphics to use. If you have only 5K, you would theoretically be limited to the first 3 modes. But because the HESPLOT machine language (M.L.) program is loaded in using a BASIC loader program which takes up more room, in reality, you can use only the first two on a 5K VIC. If you have a machine language monitor for the VIC, you could SAVE off the actual HESPLOT M.L. program to a tape, then always load just those 400 bytes. By doing this, you would be able to use mode 3 on a 5K VIC.

The first choice you have to make is the size of the hi-res area. On an 8K VIC, user memory is available from 1024 to 7680. If you want to use the maximum resolution available, by looking at column 3 of Table 1, you can see that the hi-res area of mode 4 starts at 4096. That means you have 3072 bytes (4096-1024) free for your BASIC program and its variables.You will also lose about 400 bytes from the 3072 for the HESPLOT M.L. program. If your program is larger than that, see if the next smaller mode will do. Mode 3 starts at 5120 and will leave 4096 free for your program and so on. In this manner, choose the mode that best accomodates both the needs of your program and hi-res resolution. Let us say you want to use mode 4.

The hi-res area for mode 4 starts at 4096. Therefore, memory from 4096 to 7680 must be protected from BASIC-

```
POKE 56,16: POKE 55,0: CLR
```

The above command is issued in the immediate mode (as opposed to in a program). The 16 was gotten from column 4 of Table 1. The rest should be entered as shown. Now you are ready to run HESPLOT.

2.0 Operations

2.1 Initial Load

Load in "HESPLOT" and RUN it on the VIC. The HESPLOT program contains the

machine language part at the end of the BASIC program. This BASIC program determines where the top of memory is, and puts the HESPLOT machine language program there, and protects it from BASIC. Two addresses will be printed on the screen, make a note of both of them. The first one is the address of VECTOR and the second, of PLOT. HESPLOT is now ready to use. You can now load in and run a BASIC program to plot a graph or whatever.

To summarize, when you want to run HESPLOT, first choose the hi-res mode you want. Then protect it from BASIC by doing the POKE to 55 and 56. The number to be POKEd into 56 is gotten from column 4 of Table 1. After doing the POKE, LOAD in HESPLOT and run it - it is then ready to use.

If you have trouble loading a program, try it a few times. Make sure that the cassette heads are clean and de-magnetized. Also try the other side of the cassette. Each side contains one set of programs, and then it is repeated again on the same side. Thus the tape contains four copies of all programs. If it still won't load, send it back to us with a brief explanation of the trouble. We will send you a replacement free. Our cassettes are professionally duplicated and we have an extremely low percentage of load problems.

2.2 HESPLOT Commands

As mentioned earlier, HESPLOT consists of two programs, VECTOR and PLOT. PLOT has three commands -

Command	Meaning
Ø	Reset point
1	Set point
2	Clear Screen

The command to PLOT is passed in location 191. So, to clear the screen:

```
POKE 191,2:SYS YYYY: REM clear hi-res screen
```

YYYY is the address of PLOT that was printed on the screen when HESPLOT was run (We will refer to the address of PLOT as YYYY and the address of VECTOR, as XXXX). To use either command Ø or 1, you must also pass it the X and Y coordinates of the point you want to reset/set. The X-coordinate is passed in 180, and Y-coordinate, in 181. Therefore to set the point at (10,12) on:

```
POKE 180,10:POKE 181,12: POKE 191,1: SYS YYYY
```

To reset that point, POKE Ø into 191 instead of 1.
VECTOR is the second program in HESPLOT. It will draw a straight line between two points or erase an existing line.

Command	Meaning
Ø	Erase line
1	Draw line

Two sets of points must be passed to VECTOR. The X and Y coordinates of the starting point are passed in 180 and 181, respectively and the X and Y coordinates of the ending point, in 182 and 183, respectively. To draw

a line from the origin (upper-left corner) to 100,50:

```
POKE180,0:POKE181,0:POKE182,100:POKE183,50:POKE191,1:SYS XXXX
```

If you wanted to erase that line, just POKE 0 into 191 instead of 1. One other feature of VECTOR - when it is finished, the ending point becomes the starting point, i.e., the contents of locations 182 and 183 are moved to location 180 and 181, respectively. Thus, if you want to draw a series of connected lines, after the first time, you need only poke the ending points. To give an example, let's say you want to draw a box, such as shown below. The four corners are labeled A,B,C, and D and their coordinates are also shown:

```
(10,5) A+----------+D (20,5)
       1          1
       1          1
(10,8) B+----------+C (20,8)
```

```
1. POKE 180,10:POKE 181,5:POKE 182,10:POKE 183,8:POKE 191,1:SYS XXXX
2. POKE 182,20:POKE 183,8:SYS XXXX
3. POKE 182,20:POKE 183,5:SYS XXXX
4. POKE 182,10:POKE 183,5:SYS XXXX
```

The first statement would draw a line from A to B, the second from B to C, the third from C to D and the fourth from D to A. Note that the command in location 191 also stays unchanged.

One final point about using HESPLOT must be noted. In any program that uses it, you must include the following two lines of BASIC at the beginning. This is done to set up the hi-res mode. Both VECTOR and PLOT assume that the VIC is already in the hi-res mode, so the two lines must be executed before calling them.

```
20 POKE36867,PEEK(36867)OR1:POKE36869,252
40 FORJ=0TO32*((255-PEEK(36869))*2+1)-1:POKE7680+J,J:
   POKE38400+J,PEEK(646):NEXT
```

The first line sets up the hi-res mode in the VIC chip. NOTE: the POKE to 36869 should be gotten from column 2 of Table 1, to set up whatever mode you like. In the example above, 252 was POKEd, which sets up mode 4 (176 by 160 pixels). The way to get out of the hi-res mode is to use RESTORE (Press RUN/STOP and while holding it down, press RESTORE key)

2.3 Demo Programs

To get you to started using HESPLOT and so that you can see exactly how some of the commands are actually used, several programs will now be given. They all use mode 4, so you must have an 8K VIC. These are actual programs that you can type in and run, after HESPLOT has been loaded in and run. The only change you will have to make is to the address of PLOT and VECTOR in line 2. The addresses for the examples below are a result of having loaded in and run HES's HESCOM program first.

Program 1

```
 1 PRINT CHR$(147): REM CLEAR REGULAR SCREEN
 2 PLOT = 3414
10 REM ** ARCHIMEDES SPIRAL *** MODIFIED FROM MTU AD
15 REM ALWAYS EXECUTE LINES 20-40 TO SET UP THE HI-RES MODE
20 POKE 36867,PEEK(36867)OR1:POKE36869,252
40 FORJ=0TO32*((255-PEEK(36869))*2+1)-1:POKE7680+J,J:
   POKE38400+J,PEEK(646):NEXT
90 POKE 191,2: SYS PLOT: REM CLEAR HI-RES SCREEN
100 PI=3.14159265:P=88:Q=80:XP=79:XR=1.5*PI:YP=30:YR=1:
    ZP=35:XF=XR/XP:YF=YP/YR:ZF=XR/ZP
110 FOR ZI=-QTOQ-1:IFZI<-ZPORZI>ZPGOTO150
120 ZT=ZI*XP/ZP:ZZ=ZI:XL=INT(.5+SQR(XP*XP-ZT*ZT))
130 FORXI=-XL TO XL: XT=SQR(XI*XI+ZT*ZT)*XF:XX=XI:
    YY=(SIN(XT)+.4*SIN(3*XT))*YF
135 X1%=ABS(XX+ZZ+P):Y1%=YY-ZZ+Q
136 IF Y1%=0 GOTO 140
138 POKE 180,X1%:POKE 181,160-Y1%:POKE 191,1:SYS PLOT
140 NEXT XI
150 NEXT ZI
199 GOTO 199
```

The above program will draw a fantastic three-dimensional figure called the "archimedes spiral". You may have seen it in ads by Micro Technology Inc. It only calls PLOT. Note that the program produces thousands of points to plot and it takes about 30 minutes for the figure to be completed! But half the fun is watching it plot. Note also on line 138 - instead of plotting Y1%, 160-Y1% is plotted. That is done to invert the origin, so origin (0,0) is now at the lower-left corner of the hi-res screen. Line 199 is just a loop so that the screen will stay undisturbed after the plot is done.

Program 2

The next program will produce different types of interesting plots. First duplicate lines 1 through 90 from Program 1 again.

```
100 PI=3.14159265:Z=50: K=1: T=3: A=2: B=2
110 REM CHANGE A,B,K AND T AS YOU WISH FOR DIFFERENT EFFECTS
120 REM SOME INTERESTING ONES ARE:
133 REM T=3:A=1:B=1
134 REM T=3:A=2:B=3
135 REM T=3:A=2:B=2
136 REM T=1:A=2:B=2
137 REM T=1:A=1:B=1    (ELLIPSE)
140 FOR TH=0 TO 2*PI STEP 2*PI/180
150 R=Z*SIN(TH*T)
160 X%=K*R*COS(A*TH)+88: Y%=R*SIN(B*TH)+80
210 POKE 180,X%:POKE 181,Y%:POKE 191,1:SYS PLOT: NEXT
299 GOTO 299
```

Program 3

The following program is a trivial program, it draws random lines. Its purpose is to show how to set up the call to VECTOR. Conceivably, this program can be used to graphically demonstrate randomness at work. First

duplicate lines 1 through 90 from Program 1 again. Remember to set the address of VECTOR to that which was printed on your screen when HESPLOT was first loaded in and run.

```
  2 VECTOR=3262
110 X0%=RND(1)*175:Y0%=RND(1)*159
120 X1%=RND(1)*175:Y1%=RND(1)*159
150 POKE 180,X0%: POKE 181,Y0%
160 POKE 182,X1%: POKE 183,Y1%: POKE 191,1: SYS VECTOR
170 GOTO 110
```

Two pairs of coordinates are gotten at random and a line is drawn between them. Because 1 is POKEd into 191, lines are drawn. If you want to erase a line, POKE 191,0.

3.0Appendices

3.1 References

1. PLOT can only be used on the VIC. However, VECTOR calculates the coordinates of intermediate points when drawing a line and calls PLOT to actually plot that point. If you change the PLOT routine to plot on a PET, you can use VECTOR unchanged. LPLOTSRC is a routine in machine language on pages 51-54 of Nick Hampshire's "Library of PET Subroutines". It will plot on the PET, giving a resolution of 80 by 50.

2. If you want to understand more about how HESPLOT works, you will need to know about the VIC's definable characters. Dave Malmberg wrote an article about it in the premier issue of "Home and Educational Computing", on pages 11-15. That issue was included free with the July 81 COMPUTE. In addition, the PET-POURRI column written by Robert Baker in the November Kilobaud/Microcomputing also has some information.

3. Although it may be hard to get a hold of, the March 1979 issue of MICRO has an article by John Sherburne on pages 19-23 that contains many neat mathematical formulae that can be plotted. It was written for the PET, but with HESPLOT and a VIC, you can easily modify it and get much better resolution.

3.2 HESPLOT used with HESCOM

HESCOM is a package that allows communication between a PET and a VIC (or between a PET and a PET or a VIC and a VIC). It includes a cable and machine language software. If you have HESCOM, there are several additional uses for it in conjunction with HESPLOT.

First of all, as usual, you should decide upon the hi-res mode. Say you want mode 4. As you would before loading HESPLOT, protect the hi-res area by the POKE to 55 and 56. Then load in HESCOM and run it, then load in HESPLOT and run in. Now if you create a hi-res picture (say the archimedes spiral) from the demo section above, when you are done, you can transfer the image of that picture from the VIC to the PET via HESCOM and save it from the PET to a tape or disk. Then anytime you want to display the picture on the VIC, you only have to load in the picture's image from the disk, send it to the VIC, and by running the two-line BASIC program needed for all hi-res pictures, you can display it on the

VIC again. Instead of running the archimedes spiral program again for 30 minutes, you can do the whole thing in seconds!

Here are all the steps to do this. It is assumed that you have both HESCOM and HESPLOT loaded in and that the archimedes spiral program has run for 30 minutes and it is now finished. It is also assumed that you have an 8K VIC and at least a 16K PET. Hit the RESTORE key to get out of the hi-res mode. On the VIC, type:

```
POKE 247,0:POKE 248,16:POKE 249,0:POKE 250,30
POKE 251,0:POKE 252,16:POKE 253,2:SYS 0
```

On the PET, type in: SYS 0. Now the whole hi-res picture in the VIC from 4096 to 7680 will be sent to the PET. On the PET, go into the monitor (SYS 4) and type:

```
.S "0:ARCH.PIC",08,1000,1E00
```

This will save off the picture to the disk. Now whenever you want to display it on the VIC, have HESCOM in both machines, LOAD "ARCH.PIC" into the PET, transfer it to the VIC using HESCOM:

```
POKE 184,0:POKE 185,16:POKE 186,0:POKE 187,30
POKE 188,0:POKE 189,16:POKE 190,2:SYS 0
```

And on the VIC type: SYS 0. You should also load into the VIC the two lines of BASIC code (lines 20 and 40 in demo Program 1) and run it and the picture will now be shown on the VIC. For more details about the HESCOM calls, see Section 5.1 of the HESCOM User Manual.

3.3

```
LINE# LOC   CODE        LINE

00001  0000             ;PUT"@1:DRAW.ASM
00002  0000             ;*************************************************
00003  0000             ;*****************  VECTOR  **********************
00004  0000             ;*************************************************
00005  0000             ; (C) 1981 HUMAN ENGINEERED SOFTWARE
00006  0000             ; WRITTEN BY JAY BALAKRISHNAN - 8/6/81
00007  0000             ; BASED UPON A BASIC PROGRAM ON PAGES 414-416
00008  0000             ; OF THE AUGUST 81 ISSUE OF BYTE
00009  0000             ; WORKS ON ANY VIC OR PET WITH ROMS 3.0 OR 4.0
00010  0000             ; PGM IS RELOCATABLE EXCEPT FOR THE ONE JSR TO "PLOT"
00011  0000             ; REGISTERS USED - ALL
00012  0000             ; THIS PROGRAM DRAWS A LINE BETWEEN TWO
00013  0000             ; X,Y COORDINATES
00014  0000             ; *******  CALLING SEQUENCE  *******
00015  0000             ; STARTING COORDINATES ARE SPECIFIED IN ORIGX & Y
00016  0000             ; ENDING COORDINATES ARE SPECIFIED IN ENDX & Y
00017  0000             ; PLOTTING OPTION IS SPECIFIED IN PLOTOP
00018  0000             ; PLOTOP = 0  ERASE LINE
00019  0000             ; PLOTOP = 1  DRAW LINE
00020  0000                    * = 4690
00021  1252             ; ALL BYTES BELOW WORK FOR VIC OR PET (ROM 3 OR 4)
00022  1252             ORIGX  = 180    STARTING X CO-ORD
00023  1252             ORIGY  = 181    STARTING Y CO-ORD
00024  1252             ENDX   = 182    ENDING   X CO-ORD
00025  1252             ENDY   = 183    ENDING   Y CO-ORD
00026  1252             DX     = ENDX   DIFF BETWEEN X CO-ORDS (REUSE BYTE)
00027  1252             DY     = ENDY   DIFF BETWEEN Y CO-ORDS (REUSE BYTE)
00028  1252             REMAIN = 184    2-BYTE REMAINDER (HI,LO)
00029  1252             ADJX   = 187    WORK VARIABLES
00030  1252             ADJY   = 188          "
00031  1252             TEMP2  = 189          "
00032  1252             TEMP1  = 190          "
00033  1252             PLOTOP = 191    PLOTTING OPTION


00035  1252  A0 01       VECTOR LDY #1
00036  1254  84 BC              STY ADJY
00037  1256  84 BB              STY ADJX
00038  1258  88                 DEY
00039  1259  84 BE              STY TEMP1
00040  125B  84 BD              STY TEMP2
00041  125D  84 B8              STY REMAIN  REMAIN HI BYTE = 0
00042  125F  88                 DEY         Y= -1
00043  1260  A2 01              LDX #1      OFFSET, DO Y'S FIRST


00045  1262  B5 B6       SUBLOP LDA ENDX,X
00046  1264  D5 B4              CMP ORIGX,X IF ENDX/Y < ORIGX/Y
00047  1266  B0 0C              BCS POSDIF  ( >= MEANS DX/Y IS POSITIVE)
00048  1268  38                 SEC         THEN DX/Y =ORIGX/Y - ENDX/Y
00049  1269  B5 B4              LDA ORIGX,X
00050  126B  F5 B6              SBC ENDX,X
00051  126D  95 B6              STA DX,X
00052  126F  94 BB              STY ADJX,X  ALSO ADJX OR ADJY = -1
```

```
LINE# LOC   CODE        LINE

00053  1271  38                SEC
00054  1272  B0 04             BCS DOX      ALWAYS


00056  1274  F5 B4       POSDIF SBC ORIGX,X
00057  1276  95 B6              STA DX,X     DX/Y = ENDX/Y - ORIGX/Y

00059  1278  CA           DOX   DEX
00060  1279  F0 E7              BEQ SUBLOP   NOW DO THE X'S


00062  127B  C5 B7              CMP DY       IF DX < DY
00063  127D  B0 15              BCS NOXCHG   ( >= MEANS NO EXCHANGE NEEDED)
00064  127F  A6 B6              LDX DX       THEN EXCHANGE DX AND DY
00065  1281  A5 B7              LDA DY
00066  1283  85 B6              STA DX
00067  1285  86 B7              STX DY
00068  1287  C8                 INY          Y = 0
00069  1288  A5 BB              LDA ADJX
00070  128A  85 BE              STA TEMP1    TEMP1 = ADJX
00071  128C  A5 BC              LDA ADJY
00072  128E  85 BD              STA TEMP2    TEMP2 = ADJY
00073  1290  84 BB              STY ADJX     ADJX = 0
00074  1292  84 BC              STY ADJY     ADJY = 0
00075  1294  A0 01       NOXCHG LDY #1       Y WILL CONTAIN THE COUNT
00076  1296  A5 B6              LDA DX
00077  1298  4A                 LSR A        SHIFT RIGHT = DX / 2
00078  1299  85 B9              STA REMAIN+1  REMAIN=REMAIN/DX (FITS 1 BYTE)
00079  129B  10 43              BPL PLOTIT   UNCOND GO PLOT ORIGX,ORIGY


00081  129D  18           DRWLOP CLC
00082  129E  A5 B4              LDA ORIGX
00083  12A0  65 BB              ADC ADJX
00084  12A2  85 B4              STA ORIGX    ORIGX = ORIGX + ADJX
00085  12A4  18                 CLC
00086  12A5  A5 B5              LDA ORIGY
00087  12A7  65 BD              ADC TEMP2
00088  12A9  85 B5              STA ORIGY    ORIGY = ORIGY + TEMP2
00089  12AB  18                 CLC
00090  12AC  A5 B9              LDA REMAIN+1  REMAIN = REMAIN + DY
00091  12AE  65 B7              ADC DY       FIRST LO BYTE
00092  12B0  85 B9              STA REMAIN+1
00093  12B2  A5 B8              LDA REMAIN   NOW HI BYTE
00094  12B4  69 00              ADC #0
00095  12B6  85 B8              STA REMAIN
00096  12B8  C8                 INY          COUNT = COUNT + 1
00097  12B9  A5 B8              LDA REMAIN   PLOT X,Y IF REMAIN <= DX
00098  12BB  D0 08              BNE SUBREM   IF HI BYTE, THEN REMAIN > DX
00099  12BD  A5 B9              LDA REMAIN+1  COMPARE LO BYTE
00100  12BF  C5 B6              CMP DX       WITH DX
00101  12C1  90 1D              BCC PLOTIT   IF REMAIN < DX, THEN PLOT IT
00102  12C3  F0 1B              BEQ PLOTIT   IF REMAIN = DX, THEN PLOT IT
00103  12C5  38           SUBREM SEC
00104  12C6  A5 B9              LDA REMAIN+1
```

```
LINE# LOC   CODE        LINE

00105  12C8  E5 B6              SBC DX       REMAIN = REMAIN - DX
00106  12CA  85 B9              STA REMAIN+1  LO BYTE
00107  12CC  A5 B8              LDA REMAIN
00108  12CE  E9 00              SBC #0
00109  12D0  85 B8              STA REMAIN  HI BYTE
00110  12D2  18                 CLC
00111  12D3  A5 B4              LDA ORIGX
00112  12D5  65 BE              ADC TEMP1
00113  12D7  85 B4              STA ORIGX   ORIGX = ORIGX + TEMP1
00114  12D9  18                 CLC
00115  12DA  A5 B5              LDA ORIGY
00116  12DC  65 BC              ADC ADJY
00117  12DE  85 B5              STA ORIGY   ORIGY = ORIGY + ADJY
00118  12E0  20 EA 12    PLOTIT JSR PLOT
00119  12E3  C4 B6              CPY DX       IF COUNT <= DX
00120  12E5  90 B6              BCC DRWLOP   THEN LOOP
00121  12E7  F0 B4              BEQ DRWLOP          "
00122  12E9  60                 RTS          COUNT > DX, SO RETURN


00124  12EA                    ;************************************************
00125  12EA                    ;****************  PLOT  ************************
00126  12EA                    ;************************************************
00127  12EA                    ; (C) 1981 HUMAN ENGINEERED SOFTWARE
00128  12EA                    ; WRITTEN BY JAY BALAKRISHNAN - 7/18/81
00129  12EA                    ; THIS PLOT ROUTINE IS WRITTEN FOR THE VIC
00130  12EA                    ; ROUTINE IS RELOCATABLE
00131  12EA                    ; SETS OR RESETS PIXEL IN HI-RES MODE
00132  12EA                    ; DEPENDING ON INPUT PARM IN PLOTOP (PLOTTING OPTION)
00133  12EA                    ; ********  CALLING SEQUENCE   **************
00134  12EA                    ; X,Y COORDINATE IN XCORD AND YCORD, RESP.
00135  12EA                    ; ORIGIN (0,0) IS AT THE UPPER LEFT CORNER
00136  12EA                    ; PLOTTING OPTIONS SPECIFIED IN PLOTOP
00137  12EA                    OPTRES = 0       OPTION - RESET PIXEL
00138  12EA                    OPTSET = 1       OPTION - SET PIXEL
00139  12EA                    OPTCLR = 2       OPTION - CLEAR SCREEN
00140  12EA                    ; REGISTERS USED - ALL
00141  12EA                    ; REGISTERS X AND Y SAVED FIRST
00142  12EA                    ; REUSE A FEW VARIABLES TO SAVE Z-PAGE MEMORY
00143  12EA                    XCORD  = ORIGX  PARM IN - X CO-ORD (0 - 175)
00144  12EA                    YCORD  = ORIGY  PARM IN - Y CO-ORD (0 - 159)
00145  12EA                    TEMPX  = ENDX   2 BYTE TEMP VAR (HI,LO)
00146  12EA                    TEMPY  = TEMP2     2 BYTE TEMP VAR (HI,LO)
00147  12EA                    CGPTRL = 184    LO PTR TO START OF CG RAM
00148  12EA                    CGPTRH = 185    HI PTR TO START OF CG RAM
00149  12EA                    CGVAL  = 186    VALUE USED TO FIG OUT CGRPTR
00150  12EA                    YA     = YCORD  REUSE
00151  12EA                    YB     = XCORD
00152  12EA                    XA     = YA     REUSE
00153  12EA                    XB     = YB
00154  12EA                    PROD   = TEMPY  HI,LO
```

```
LINE# LOC   CODE        LINE

00156  12EA  8A          PLOT   TXA
00157  12EB  48                 PHA          SAVE REG X
00158  12EC  98                 TYA
00159  12ED  48                 PHA          SAVE REG Y
00160  12EE  A2 0B              LDX #PLOTOP-ORIGX
00161  12F0  B5 B3       SAVELP LDA ORIGX-1,X   SAVE BYTES FROM ORIGX TO PLOTOP
00162  12F2  48                 PHA          ON THE STACK AND USE THEIR SPACE
00163  12F3  CA                 DEX
00164  12F4  D0 FA              BNE SAVELP


00166  12F6                    ; CALCULATE START OF CHARACTER GENERATOR RAM
00167  12F6                    ; START OF CG RAM PTR = 28-((255-PEEK(36869))*4)
00168  12F6  38                 SEC
00169  12F7  A9 FF              LDA #255
00170  12F9  ED 05 90           SBC 36869  VIC CHIP'S CHAR GEN REGISTER
00171  12FC  0A                 ASL A      SHIFT LEFT ONCE = MULT BY 2
00172  12FD  85 BA              STA CGVAL  SAVE OFF THE DIFF * 2 FOR LATER
00173  12FF  0A                 ASL A      SHIFT ONCE MORE = MULT BY 4
00174  1300  85 B9              STA CGPTRH SAVE DIFF * 4 HERE BRIEFLY
00175  1302  38                 SEC
00176  1303  A9 1C              LDA #28
00177  1305  E5 B9              SBC CGPTRH  HI ONLY, ALWAYS ON PAGE BNDRY
00178  1307  85 B9              STA CGPTRH
00179  1309  A9 00              LDA #0
00180  130B  85 B8              STA CGPTRL BYTES (LO,HI) OF CGPTR NOW SET


00182  130D  A5 BF              LDA PLOTOP GET OPTION
00183  130F  C9 02              CMP #OPTCLR  CLEAR SCREEN?
00184  1311  D0 24              BNE GETCRD  NOPE

00186  1313                    ; *****  CLEAR SCREEN ROUTINE  *****
00187  1313                    ; CGVAL CONTAINS (255-PEEK(36869)*2)
00188  1313                    ; NUMBER OF BYTES TO CLEAR  = (CGVAL+1)*8*64
00189  1313  E6 BA       CLRSCN INC CGVAL  CGVAL = CGVAL + 1
00190  1315  06 BA              ASL CGVAL  CGVAL = CGVAL * 8  (OUTER LOOP)
00191  1317  06 BA              ASL CGVAL
00192  1319  06 BA              ASL CGVAL
00193  131B  A0 3F       CLRLP1 LDY #63     INNER LOOP COUNTER (64 TIMES)
00194  131D  A9 00              LDA #0
00195  131F  91 B8       CLRLP2 STA (CGPTRL),Y  PUT 0 HERE, CLEARING SCREEN
00196  1321  88                 DEY
00197  1322  10 FB              BPL CLRLP2
00198  1324  C6 BA              DEC CGVAL
00199  1326  F0 56              BEQ DONE   DONE IF CGVAL = 0, GO LONG WAY
00200  1328  18                 CLC
00201  1329  A5 B8              LDA CGPTRL
00202  132B  69 40              ADC #64    CGPTR = CGPTR + 64
00203  132D  85 B8              STA CGPTRL LO BYTE
00204  132F  A5 B9              LDA CGPTRH
00205  1331  69 00              ADC #0
00206  1333  85 B9              STA CGPTRH HI BYTE
00207  1335  90 E4              BCC CLRLP1 ALWAYS
```

```
LINE# LOC    CODE        LINE

00209  1337  A4 B5     GETCRD LDY YCORD  GET Y CO-ORD
00210  1339  A6 B4            LDX XCORD  GET X CO-ORD

00212  133B             ; GET Y-VALUE

00214  133B  98                 TYA
00215  133C  4A                 LSR A       SHIFT RIGHT 4 TIMES
00216  133D  4A                 LSR A
00217  133E  4A                 LSR A
00218  133F  4A                 LSR A
00219  1340  85 B5              STA YA      YA = Y/16
00220  1342  98                 TYA         GET 4 LOW ORDER BITS
00221  1343  29 0F              AND #%00001111  YB = Y-YA*16
00222  1345  0A                 ASL A       SHIFT YB LEFT 3 TIMES
00223  1346  0A                 ASL A
00224  1347  0A                 ASL A
00225  1348  A8                 TAY         SAVE YB IN Y FOR NOW
00226  1349  8A                 TXA
00227  134A  48                 PHA         SAVE X CORD, MULT RTN USES X REG


00229  134B             ; MULTIPLICATION ROUTINE
00230  134B             ; PROD = 2816 * YA
00231  134B             ; PRECISION : 16 BITS * 8 BITS = 16 BITS

00233  134B  A9 00              LDA #0
00234  134D  85 BD              STA PROD
00235  134F  85 BE              STA PROD+1  ZERO OUT RESULT FIELD
00236  1351  A5 B5              LDA YA      GET MULIPLIER
00237  1353  A2 08              LDX #8      MULTIPLIER HAS 8 BITS
00238  1355  18          LOOP0  CLC
00239  1356  26 BE              ROL PROD+1  SHIFT PARTIAL RESULT LEFT
00240  1358  26 BD              ROL PROD    2-BYTE SHIFT
00241  135A  0A                 ASL A       SHIFT MULTIPLIER BIT OUT
00242  135B  90 0F              BCC NOCARY
00243  135D  48                 PHA         SAVE MULTIPLIER (YA)
00244  135E  A5 BE              LDA PROD+1  GET RESULT LO BYTE
00245  1360  18                 CLC
00246  1361  69 00              ADC #0      LO BYTE OF $0B00 (DEC. 2816)
00247  1363  85 BE              STA PROD+1
00248  1365  A5 BD              LDA PROD    RESULT HI BYTE
00249  1367  69 0B              ADC #$0B    HI BYTE OF $0B00
00250  1369  85 BD              STA PROD
00251  136B  68                 PLA         GET BACK MULTIPLIER
00252  136C  CA          NOCARY DEX
00253  136D  D0 E6              BNE LOOP0   LOOP 8 TIMES

00255  136F  18                 CLC
00256  1370  98                 TYA         GET YB FROM REG Y
00257  1371  65 BE              ADC PROD+1  AND ADD YB TO PROD
00258  1373  85 BE              STA TEMPY+1  SAVE LO ORDER BYTE
00259  1375  A9 00              LDA #0
00260  1377  65 BD              ADC PROD
00261  1379  85 BD              STA TEMPY   SAVE HI BYTE
```

```
LINE# LOC   CODE        LINE

00263  137B  38              SEC
00264  137C  B0 02           BCS GETX    ALWAYS; GOTO TO SKIP OVER 'DONE'


00266  137E                  ; NEXT BRANCH IS IN MIDDLE OF PGM SO BRANCHES FROM
00267  137E                  ; BEG. OF PGM CAN REACH 'DUN' AND STAY RELOCATABLE
00268  137E  F0 69           DONE   BEQ DUN     Z-FLAG SHOULD BE SET ON

00270  1380                  ; GET X-VALUE

00272  1380  68              GETX   PLA         GET SAVED X CORD
00273  1381  AA                     TAX         GET 3 LO-ORDER BITS
00274  1382  29 07                  AND #%00000111  OF X CORD
00275  1384  85 B4                  STA XB      XB = X-8*XA
00276  1386  8A                     TXA         GET X CORD AGAIN
00277  1387  4A                     LSR A       SHIFT RIGHT 3 TIMES
00278  1388  4A                     LSR A
00279  1389  4A                     LSR A       REG A = XA = X/8
00280  138A  A2 00                  LDX #0
00281  138C  86 B6                  STX TEMPX   HI BYTE TO 0
00282  138E  A2 07                  LDX #7      SHIFT LEFT 7 TIMES
00283  1390  0A              LOOP1  ASL A       TEMPX = XA * 128
00284  1391  26 B6                  ROL TEMPX   TAKE BITS COMING OFF .A
00285  1393                  ;                     AND PUT INTO TEMPX
00286  1393  CA                     DEX
00287  1394  D0 FA                  BNE LOOP1   DO 7 TIMES
00288  1396  18                     CLC         ADD XB TO TEMPX
00289  1397  65 B4                  ADC XB      ADD TO REG A (LO BYTE, XA*128)
00290  1399  85 B7                  STA TEMPX+1  LO BYTE
00291  139B  A9 00                  LDA #0
00292  139D  65 B6                  ADC TEMPX
00293  139F  85 B6                  STA TEMPX   HI BYTE

00295  13A1                  ; ADD TEMPX TO TEMPY = RELATIVE PIXEL # (0-30975)

00297  13A1  18                     CLC
00298  13A2  A5 BE                  LDA TEMPY+1 LO BYTES 1ST
00299  13A4  65 B7                  ADC TEMPX+1
00300  13A6  85 BE                  STA TEMPY+1
00301  13A8  85 B7                  STA TEMPX+1 SAVE LO BYTE HERE TOO
00302  13AA  A5 BD                  LDA TEMPY   NOW HI BYTES
00303  13AC  65 B6                  ADC TEMPX
00304  13AE  85 BD                  STA TEMPY
00305  13B0  A2 03                  LDX #3      SHIFT TEMPY RIGHT 3 TIMES
00306  13B2  46 BD           LOOP2  LSR TEMPY   TEMPY = TEMPY/8
00307  13B4  66 BE                  ROR TEMPY+1 (2 BYTE SHIFT)
00308  13B6  CA                     DEX
00309  13B7  D0 F9                  BNE LOOP2   AT END,TEMPY = REL BYTE #(HI,LO)
00310  13B9  A5 B7                  LDA TEMPX+1  GET SAVED TEMPY+1 (LO)
00311  13BB  29 07                  AND #%00000111  AND GET THE LO 3 BITS
00312  13BD  85 B7                  STA TEMPX+1  STORE IT, THE BIT # WITHIN BYTE


00314  13BF                  ; TEMPY NOW CONTAINS THE RELATIVE BYTE (HI,LO)
00315  13BF                  ; AND TEMPX+1 THE BIT WITHIN THAT BYTE
```

```
LINE# LOC    CODE          LINE

00316  13BF                ; SO NOW ACTUALLY SET THE PIXEL ON OR OFF

00318  13BF  A6 BD                LDX TEMPY     GET HI BYTE AND HOLD FOR NOW
00319  13C1  18                  CLC           TURN REL. BYTE INTO ABS. BYTE
00320  13C2  A5 BE                LDA TEMPY+1   ADD REL BYTE
00321  13C4  65 B8                ADC CGPTRL    AND BEG. OF CG RAM, LO BYTES
00322  13C6  85 BD                STA TEMPY     STORE IN LO,HI FORMAT
00323  13C8  8A                   TXA           ADD HI BYTE OF REL BYTE
00324  13C9  65 B9                ADC CGPTRH    AND HI BYTE OF BEG OF CG RAM
00325  13CB  85 BE                STA TEMPY+1   STORE INTO HI OF (LO,HI) FORMAT


00327  13CD                ; TEMPY NOW CONTAINS LO,HI BYTES OF EXACT CG RAM BYTE
00328  13CD                ; SO NOW SET BIT AS SAVED IN TEMPX+1 OF BYTE AT TEMPY
00329  13CD                ; FOR PLOTTING, BITS ARE NUMBERED LEFT TO RIGHT
00330  13CD                ; INSTEAD OF THE NORMAL RIGHT TO LEFT

00332  13CD  A9 80                LDA #%10000000 START WITH MASK FOR BIT 0
00333  13CF  A6 B7                LDX TEMPX+1   CONTAINS THE BIT # (0-7)
00334  13D1  F0 04                BEQ SETBIT    BIT FOUND, SET/RESET BIT
00335  13D3  4A           GETBIT  LSR A         SHIFT MASK RIGHT
00336  13D4  CA                   DEX
00337  13D5  D0 FC                BNE GETBIT    LOOP TILL THE MASK IS CORRECT
00338  13D7                ;                       FOR THAT PARTICULAR BIT


00340  13D7  A4 BF        SETBIT  LDY PLOTOP    GET ACTUAL OPTION
00341  13D9  C0 00                CPY #OPTRES   CHECK OUT OPTION
00342  13DB  F0 06                BEQ BITOFF
00343  13DD  01 BD                ORA (TEMPY,X) SET BIT ON, X ALWAYS 0
00344  13DF  81 BD                STA (TEMPY,X)
00345  13E1  D0 06                BNE DUN       ALWAYS TRUE
00346  13E3  49 FF        BITOFF  EOR #$FF      COMPLEMENT .A
00347  13E5  21 BD                AND (TEMPY,X) SET BIT OFF, X ALWAYS 0
00348  13E7  81 BD                STA (TEMPY,X)
00349  13E9  A2 00        DUN     LDX #0        RESTORE ALL SAVED BYTES
00350  13EB  68           RSTRLP  PLA           FROM ORIGX TO PLOTOP
00351  13EC  95 B4                STA ORIGX,X   IN REVERSE ORDER
00352  13EE  E8                   INX
00353  13EF  E0 0B                CPX #PLOTOP-ORIGX
00354  13F1  90 F8                BCC RSTRLP
00355  13F3  68                   PLA
00356  13F4  A8                   TAY           RESTORE Y
00357  13F5  68                   PLA
00358  13F6  AA                   TAX           RESTORE X
00359  13F7  60                   RTS
00360  13F8                       .END


ERRORS = 00000
```